CDS/6E20/02AA

Combinational Logic Synthesis Research Report for Advanced Logic Synthesis for Low Power Mobile Applications Project

Alex Saldanha
Viorica Simion

Cadence Design Systems, Inc.
Cadence-Berkeley Labs
**555 River Oaks Parkway**
San Jose, CA 95134

August, 1997

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

This report and its contents may be distributed without restriction.

DTIC QUALITY INSPECTED 8
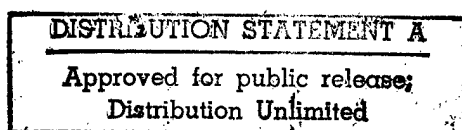
19970923 001

Evaluation of Timed Shannon Circuits in Logic Optimization

Alexander Saldanha and Viorica Simion

Cadence Design Systems, Inc.

Timed Shannon Circuits have been proposed as a low-power circuit design style [1] with the attractive properties of providing predictable, delay-insensitive low-power dissipation. In this report we present the results of a comprehensive evaluation to compare the designs generated using Timed Shannon Circuits versus those generated by a commercial logic synthesis program (Synergy).

1.0 Introduction

Timed Shannon Circuits (TSC's) have been proposed as a design alternative for low-power combinational logic circuits [1]. A Timed Shannon Circuit has the following attractive properties: (1) Power is minimized by ensuring that minimum transition activity and no glitches occur when combinational logic evaluates. (2) The power consumed is independent of the delay of the circuit - thus accurate analysis and synthesis optimizations are facilitated.

A collection of algorithms to support the generation of good quality TSC's is described in detail in [1] and a prototype implementation has been performed in the SIS logic synthesis system from U.C. Berkeley. Preliminary results on 38 well-known and standard benchmark circuits demonstrated the potential of TSC's for power minimization at some expense in area and/or delay in most of the circuits. Based on the results in [1] we have attempted to deploy the Timed Shannon Circuit design style in a commercially available logic synthesis and optimization tool, namely Synergy from Cadence Design Systems, Inc. The goal of this work is to determine the feasibility of using TSCs in a design scenario where it is recognized that besides power consumption, constraints for delay and area (as well as testability and other metrics) must be satisfied by any implementation.

This report is organized as follows. Section 2.0 is a brief review of Timed Shannon Circuits and their operation. The reader is referred to [1] for additional details of TSC's. Section 3.0 describes the power estimation used by Synergy to measure the data for the experiment. The logic optimization steps utilizing TSC's are discussed in Section 4.0. Experimental results comparing TSC's versus conventional logic optimization for area and delay optimization modes are presented and analyzed in Section 5.0. Related work that impacts the direction of future work in power optimization for combinational logic circuits is discussed in Section 6.0. Conclusions are summarized in Section 7.0.

2.0 Timed Shannon Circuits

2.1 Construction of a TSC

There are three main steps in deriving a TSC implementation for a combinational logic circuit.

First, an initial TSC is derived from the BDD of the Boolean function representing a circuit. Figure 1 provides an example of this construction. The main feature of the initial TSC implementation is that at most one path from the root (**Enable** signal in the circuit) to the terminal (output in the circuit) propagates a transition to evaluate the output value for a given vector.

The second step is composed of the steps of decomposition of high fanin gates and area recovery steps to alleviate some of the penalty imposed by the BDD's. The decomposition and area recovery steps are described in [1] and are incorporated in the prototype implementation.

After the first two steps, the TSC ensures that the power dissipation is minimal within the internal gates of the circuit. However, the fanout on the primary inputs of the TSC may be very high (the fanout of a PI is equal to the number of edges crossing at the level of the corresponding variable in the BDD), and this accounts for a substantial amount of the power dissipation in TSC's [1]. In [1] two approaches were suggested to trade-off the high fanout on primary inputs versus additional transitions within the circuit. These optimizations constitute the third step of the TSC derivation in the prototype implementation.

Note that in all three steps, an exact power analysis can be performed if the switching activity on the primary inputs is provided and the primary inputs are assumed independent. For the purposes of this experiment we use a switching probability of 0.5 on each primary input and also assume independence. Related work on a

more appealing approach that accounts for input correlations as well as better accounting of the input activity has been performed in [9] and is attached with this report.

FIGURE 1. Construction of the Timed Shannon Circuit for the parity function
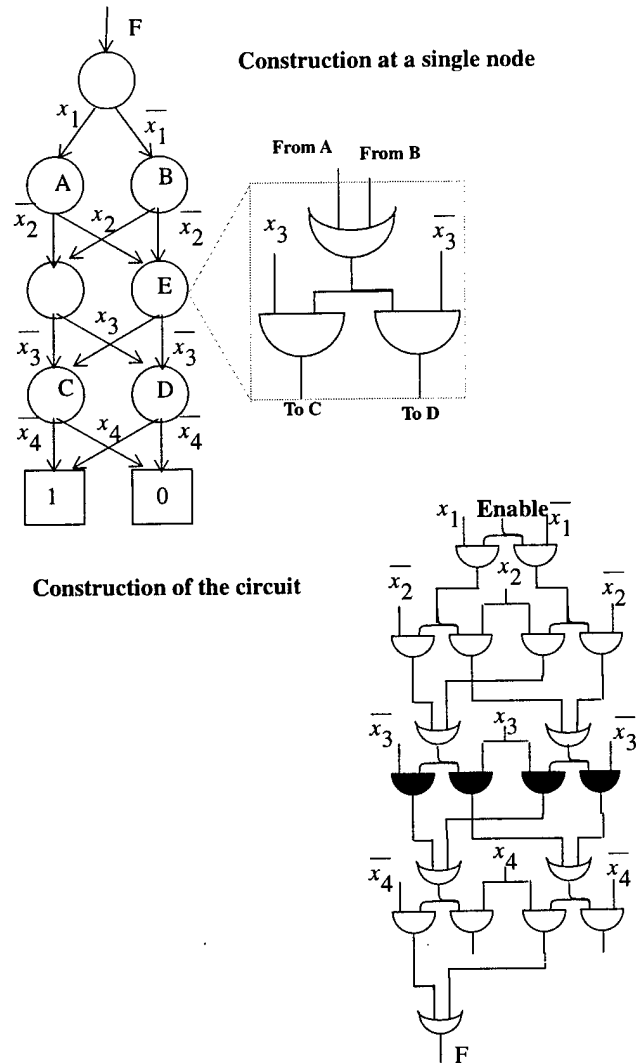
2.2 Operation of a TSC

The TSC is designed to be operated in a clocked mode, using the **Enable** signal to operate the timing scheme. The **Enable** signal is first set to 0 so that all nodes in the circuit evaluate to 0. Next the circuit inputs are changed. Since all gates are at 0 and each input is connected to an **AND** gate, whose other input is at 0, there are no transitions within the circuit. After the circuit inputs settle to their new values, the **Enable** signal is set to 1, and precisely those nodes on the single selected path from the root to a terminal node are set to 1. The value is then read from the output terminal of the circuit, the **Enable** line is set to 0 again, and the cycle is repeated.

A timing diagram comparing the operation of a TSC to a normal (non-TSC) circuit is shown below. Note that the TSC requires a two-phase clocking scheme - primary inputs change on the falling edge of the clock and primary outputs are sampled on the rising edge of the clock. The **Enable** signal can be derived in straightforward

fashion from the **Clock** signal. Each evaluation of a TSC may be described as a sequence of four steps illustrated in the figure.

From the timing diagram it is clear that the duration of the first (level 0) clock phase for the TSC is determined by the time taken for the inputs to settle after they are changed plus the time for the circuit outputs to settle once the enable signal has been set to 1 from 0. This is equal to the length of the longest path in the TSC plus the time to allow the input signals to settle on a change. For the normal circuit, the clock period is determined by the length of the longest path delay. For comparison purposes we compare only the longest path delays of the TSC with those of the normal circuit.

Note that the current implementation of TSC's requires the circuit be reset on every vector. Thus if the majority of input values do not change from vector to vector there may be substantially larger power consumed by the TSC compared to a normal implementation. Although techniques to mitigate this have been suggested in [1], they are not utilized in the prototype implementation reported here.

FIGURE 2.　Timing operation of Timed Shannon Circuits
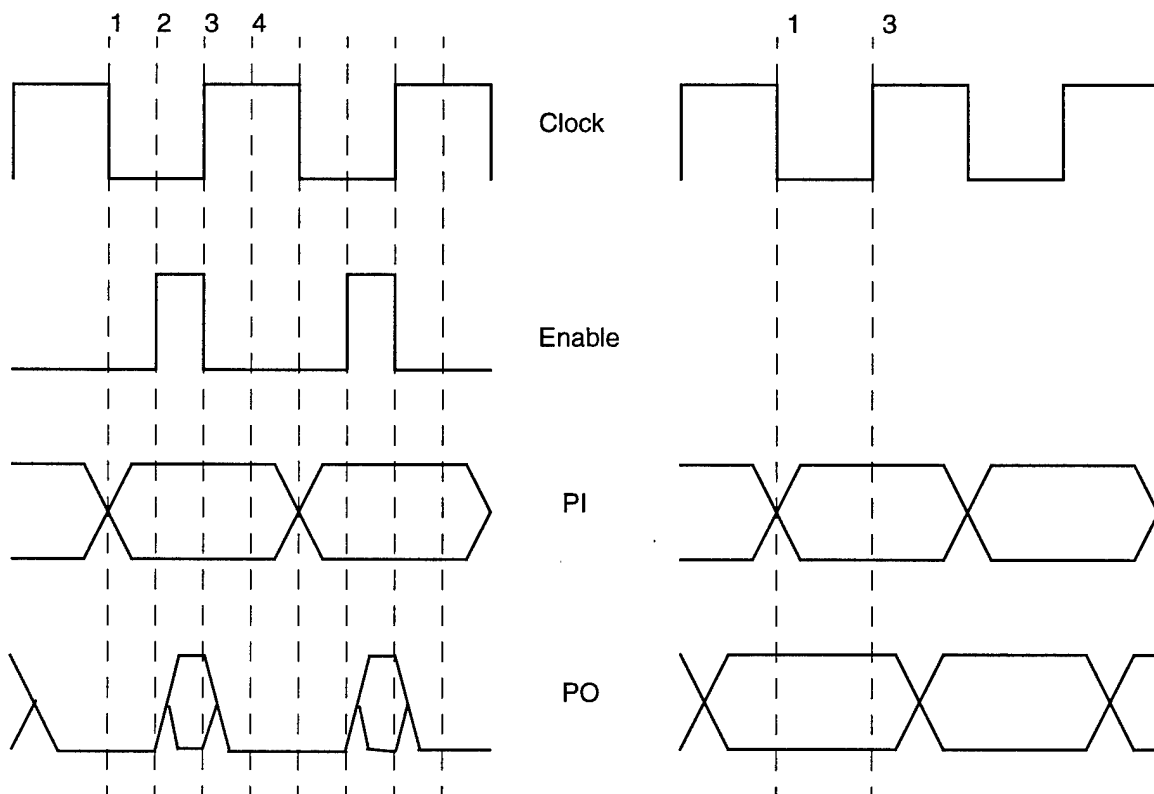
2-Phase Timed Shannon Circuit Operation
1: Primary inputs changed
2: Primary inputs settled - enable evaluation (Enable = 1) for TSC
3: Primary Outputs sampled - disable evaluation (Enable = 0) for TSC
4: All gates settled at value 0 (reset)

2-Phase Normal Circuit Operation

3.0 Power estimation in Synergy

The power estimation in Synergy is integrated together with the optimization flow. First, the logic-level description of the design is synthesized and optimized for area or timing and then the gate-level power analysis is performed. Currently, only the power analysis for combinational circuits is supported. Synergy design flow and the power estimation procedure for combinational circuits are briefly discussed in the following sections.

3.1 Design flow

The input for Synergy is a logic-level design description as shown in Figure 3. The combinational logic blocks are synthesized into an optimal netlist of gates at a Technology Independent (TI) level followed by a technology mapping step and a Technology Dependent (TD) optimization process.

The two optimization criteria accepted by Synergy are *area* and *timing*. Running a design in *area mode* means that the focus of the algorithms applied during the optimization steps is to minimize the area occupied by the logic gates and interconnect, while running a design in *timing mode* means that the function to be minimized is the critical path delay.

TI optimization step derives an optimized structure for the circuit independent of the gates available in a particular technology library. The techniques applied at this level for area optimization are usually the node extraction, simplification, and elimination. The final result is dependent on the starting circuit and the order in which these operations are performed. The typical TI operations for timing optimization are Extract, Simplify, Collapse, and Eliminate. The key to a good representation is to accurately predict the effect of each transformation, therefore, a good delay estimator is required. The output of TI optimization process is an optimized Boolean network. More details about TI optimization techniques are given in [6], [7], [8].

FIGURE 3.   Design flow in Synergy.
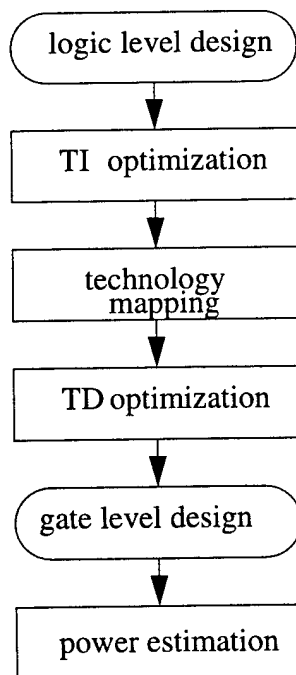
After TI optimization phase is completed, the technology mapper is computing a network of gates of minimum cost equivalent to the given Boolean network. The technology mapping transformation implies two distinct operations: recognizing logic equivalence between two logic functions (*matching* operation), and finding the best set of logically equivalent gates (*covering* operation) whose interconnection represent the original circuit. The quality of the final implementation depends significantly on the initially provided Boolean network.

The last optimization step is performed at technology dependent level on a mapped circuit. It includes gate sizing (selecting from a set of given functionally equivalent gates), fanout optimization (duplicating a gate to reduce its fanout load or buffer insertion), and fixing maximum fanout and maximum transition violations. The result is an optimized gate-level design in a target technology.

3.2 Gate-level power estimation

The power estimation in Synergy is performed on a mapped netlist as shown in Figure 3. Under a non-linear delay model assumption the total amount of power drawn from the power supply by a CMOS gate is summarized by:

$$P_{gate} = P_{fun} + P_{glitch} + P_{sc} + P_{leak} \qquad (1)$$

where P_{fun}, denoted as *functional power dissipation*, is the power required to charge or discharge the gate output capacitance in order to perform a computational task; P_{glitch}, denoted as *glitch power dissipation*, is due to the multiple transitions within one clock cycle until the output of the gate is stabilized; P_{sc} is the power dissipated during output transitions due to the current flowing from the supply to ground denoted as *short-circuit power dissipation*; and finally P_{leak} represents the *static power dissipation* due to the leakage current. The power dissipation components are briefly discussed here.

Capacitive power dissipation. The dominant source of power dissipation in CMOS circuits is the charging and discharging of the node capacitances. This sort of power dissipation, also referred as the *capacitive power dissipation*, is the sum of functional and glitch power dissipation. It is given by:

$$P_{cap} = 0.5\alpha C_L V_{DD}^2 f_{clk} \qquad (2)$$

where C_L is the physical capacitance at the output of the gate, V_{DD} is the supply voltage, f_{clk} is the clock frequency, and α (referred to as the *switching activity*) is the average number of output transitions per clock period ($1/f_{clk}$) time. Of those factors, V_{DD} and f_{clk} are design known parameters, while C_L and α have to be determined.

The **physical capacitance** C_L accounts for the input capacitance of all the gates in the fanout of a particular node, the interconnect, and the physical output capacitance of the driving gate itself.

Calculation of **switching activity** α depends on [2]:

*   input patterns and the sequence in which they are applied
*   delay model used
*   circuit structure

Switching activity at the output of a gate depends not only on the switching activity at the inputs of the gate and the logic function of the gate, but also on the spatial and temporal dependencies among the gate inputs.

Depending upon the method used to generate the switching activity, these multiple dependencies may or may not be taken into account.

In Synergy the switching activity calculation is based on **circuit simulation**. The main advantage of this technique is that existing simulators can be used, and issues such as glitch generation and propagation, and signal correlation are automatically taken into consideration. Cadence Verilog-XL based gate-level simulation program was adapted to report the switching activity per gate under random generated input sequences. The simulation techniques rely on the macromodels built for the gates in the ASIC library, as well as on the detailed gate-level timing analysis. The accuracy of the results depends on the quality of the macromodels, the glitch-filtering scheme used and the accuracy of physical capacitances provided at the gate level. In our experiments we used commercial libraries and non-linear delay models to assure quality results.

The only source of inaccuracy left in our approach of estimating power by using a simulator is the input pattern-dependence problem. Randomly generated input sequences tend to introduce estimation error. For real circuits the switching activity at the primary inputs might follow a certain pattern and the input signals might be correlated. Our experiments, however, were done on MCNC benchmark suite [5] for which the input pattern information was not available. Therefore, the input sequences were randomly generated in a sufficient high number to reduce the estimation error.

Short-circuit power dissipation. The short-circuit power consumption is due to the current flowing from the supply to the ground during an output transition. It is proportional to the input slope of the gate, the output load capacitance, and the transistor sizes of the gate. The maximum short-circuit current flows when there is no load. This current decreases with the load but increases with the input slope. For ASIC designs, the libraries are pre-characterized for short-circuit power dissipation [3].

Static power dissipation. The static power dissipation refers to the sum of leakage and standby dissipations. Leakage currents depend on the device technology, while the standby currents depend on the design logic style. For CMOS design style the standby dissipation is insignificant [4].

Total power dissipation. Since both short-circuit and static power dissipation are technology and library dependent parameters and they were not available in the ASIC library characterization, Synergy does not yet account for them. Therefore, the total power dissipated by a circuit is calculated with the following relation:

$$P_{total} = \sum_{i=1}^{m} P_{cap}(g_i) + P_{PI} \tag{3}$$

where g_i is a gate in the circuit, P_{cap} is the capacitive power dissipated by the gate, and m is the total number of gates in the circuit. The capacitive power dissipation is calculated with the relation (2) in which the average switching activity α is determined based on circuit simulation. To the power dissipated by the gates in the circuit we added the power dissipated in charging and discharging of the primary inputs, referred as P_{PI}.

The power estimation flow implemented in Synergy is presented in Figure 4. First, the gate-level design description is translated into a verilog netlist (referred as *design.v*), then the verilog test file, *test.v*, is generated and the Verilog-XL simulator is started. Another necessary input for the simulator is the Verilog ASIC library file, *lib.v*, which contains information on gate models, non-linear delays, and wire load models. The simulator generates the switching activity file, *design.switch*, based on circuit structure, primary inputs test vectors, and library information. Finally, the total capacitive power dissipated by the circuit is estimated with the relation (3).

FIGURE 4.   Power estimation flow.
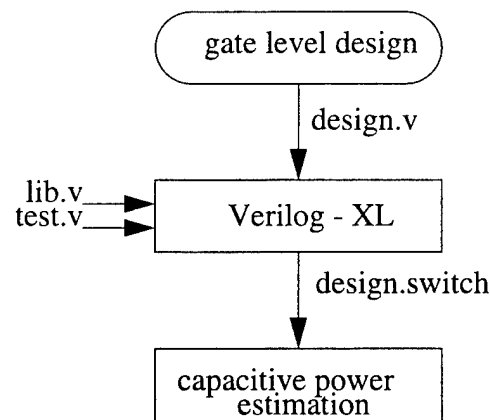
4.0   Power Optimization in Synergy using Timed Shannon Circuits

Two distinct optimization flows are compared for power, area, and delay performance. The first flow is the TSC flow in which the circuits are optimized for power, delay, and area. The second flow is the standard Synergy flow in which only the area and timing optimization are performed. In Figure 5 these two flows are presented. The MCNC benchmark circuits [5] are used in comparing the power, area, and delay performance of these two optimization flows.

TSC flow. In Timed Shannon Circuits the power optimization is performed at the technology independent level. Therefore, when integrating TSC in the Synergy flow, TI optimization techniques are not applied to these power optimized circuits. The MCNC circuit, denoted *design.blif*, is first optimized for power using the TSC method. The resulted circuit (*design.opt.blif*) is then mapped in a given technology library and optimized for area or timing at the technology dependent level using the methods briefly presented in Section 3.1. Finally, the power analysis is performed and the power, area, and timing results are reported.

Standard flow. The circuits synthesized and optimized with the standard Synergy flow are following all of the steps presented in Section 3.1. At the end of the optimization process, the gate-level design is analyzed for power and the results are compared with TSC results.

FIGURE 5.   TSC flow and standard Synergy flow.
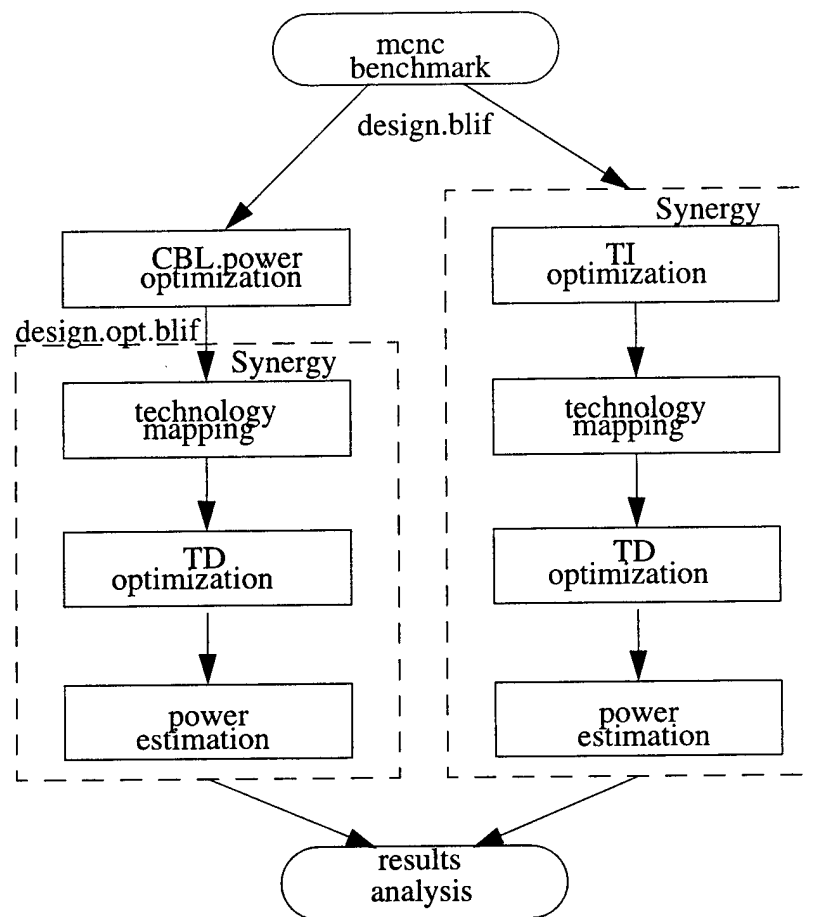
5.0 Results and analysis

TABLE 1. Area-mode results of low-power optimization

	Name	In	Out	Synergy				Shannon Circuit			
				Gate Count	Area	Delay	Power	Gate Count	Area	Delay	Power
1	5xp1	7	10	38	143	8.28	19.48	76	311	7.08	38.02
2	9sym	9	1	25	101	5.34	10.65	44	167	7.50	23.76
3	alu4	14	8	55	211	5.60	26.54	656	2605	21.57	104.35
4	apex1	45	45	500	1925	15.23	100.63	1199	4529	27.60	102.19
5	apex3	54	50	548	2218	13.82	109.73	1376	5288	17.42	121.86
6	apex4	9	19	853	3559	16.77	147.87	2176	8600	19.63	156.26
7	apex5	117	88	319	1291	10.85	116.00	960	3889	37.81	186.37
8	b12	15	9	38	135	3.19	15.32	70	268	5.72	33.85
9	bw	5	28	80	312	16.36	32.99	112	416	7.34	40.66
10	clip	9	5	43	165	5.91	19.90	118	498	11.51	45.46
11	con1	7	2	11	40	2.30	5.67	20	69	4.18	12.14
12	cordic	23	2	27	115	7.72	15.27	91	355	20.82	26.58
13	cps	24	109	399	1480	15.74	69.37	743	2866	19.95	55.54
14	duke2	22	29	159	621	8.16	36.67	361	1379	15.65	29.46
15	e64	65	65	161	512	23.61	43.46	192	604	38.57	57.58
16	ex4	128	28	190	815	7.93	95.80	548	2103	27.69	164.19
17	ex5	8	63	177	668	10.65	51.23	429	1620	12.20	151.86
18	inc	7	9	48	193	11.63	17.72	73	287	6.96	22.30
19	misex1	8	7	27	102	5.92	11.94	34	134	5.10	14.56
20	misex2	25	18	51	187	5.48	17.11	68	241	5.44	20.76
21	misex3	14	14	249	1023	16.40	82.60	1023	4024	21.56	149.89
22	misex3c	14	14	212	866	10.08	69.79	310	1304	13.97	105.37
23	pdc	16	40	232	886	9.00	55.43	240	919	11.98	50.75
24	rd53	5	3	17	62	3.31	7.27	31	120	5.31	19.72
25	rd73	7	3	32	121	7.52	14.33	51	208	5.91	29.36
26	rd84	8	4	44	177	9.95	22.95	69	285	6.49	33.28
27	sao2	10	4	49	199	11.09	20.17	177	663	14.04	23.37
28	seq	41	35	404	1537	14.37	78.49	962	3730	25.44	46.46
29	squar5	5	8	28	108	5.26	13.23	38	150	5.11	21.19
30	t481	16	1	25	79	3.80	10.95	78	312	13.30	31.81
31	table5	17	15	309	1232	13.41	57.05	757	3022	24.82	51.88
32	vg2	25	8	45	168	7.31	20.38	56	234	10.48	29.17
33	xor5	5	1	8	28	3.78	5.48	9	49	3.69	7.89
34	Z5xp1	7	10	34	135	7.29	17.01	80	318	6.84	37.15
35	Z9sym	9	1	25	101	5.34	10.65	44	167	7.50	23.74

TABLE 2. Timing-mode results of low-power optimization

	Name	Synergy				Shannon Circuit			
		Gate Count	Area	Delay	Power	Gate Count	Area	Delay	Power
1	5xp1	102	494	2.33	45.52	138	664	4.02	61.59
2	9sym	71	330	2.45	25.45	77	363	5.36	41.38
3	alu4	513	2678	4.26	227.68	970	4211	10.34	132.68
4	apex1	911	3986	6.03	213.74	1493	5895	13.15	128.31
5	apex3	765	3902	6.23	290.47	1911	8007	8.69	178.42
6	apex4	1160	6031	5.75	384.42	2689	11169	10.05	220.14
7	apex5	939	4305	3.17	303.55	1269	5110	15.44	206.41
8	b12	62	215	2.37	21.70	119	575	3.17	53.49
9	bw	152	631	2.30	50.95	173	738	3.54	64.18
10	clip	155	801	2.55	69.83	199	966	5.58	73.10
11	con1	19	118	0.91	11.02	35	175	1.94	21.48
12	cordic	94	546	3.12	49.07	172	944	12.11	46.85
13	cps	998	4825	3.58	335.49	1055	4229	11.40	70.85
14	duke2	284	1384	3.42	78.35	521	2291	7.80	41.40
15	e64	243	1081	3.29	80.74	228	960	31.15	60.86
16	ex4	312	1405	3.08	146.48	779	3155	15.55	195.14
17	ex5	212	1174	2.81	103.12	563	2370	6.52	188.41
18	inc	72	344	2.50	32.11	125	612	3.28	34.74
19	misex1	47	246	1.85	21.73	62	303	2.75	25.95
20	misex2	115	570	1.80	42.96	96	418	3.76	28.93
21	misex3	549	2766	4.62	221.50	1355	5700	11.92	200.53
22	misex3c	315	1467	4.78	126.61	439	2029	7.02	138.51
23	pdc	447	2241	8.05	216.17	331	1412	6.19	69.22
24	rd53	51	290	1.90	25.98	57	286	2.56	37.22
25	rd73	114	548	2.62	46.64	84	431	4.18	49.80
26	rd84	178	879	3.66	75.72	134	723	4.17	70.55
27	sao2	122	620	2.74	42.35	295	1311	7.70	38.61
28	seq	1237	5686	5.49	345.17	1334	5473	12.23	62.55
29	squar5	47	255	1.35	22.69	62	298	2.36	31.39
30	t481	39	301	1.79	23.36	147	818	6.54	49.90
31	table5	649	3072	7.27	148.07	988	4245	13.66	72.68
32	vg2	84	389	2.16	35.93	83	410	5.94	42.63
33	xor5	22	133	1.56	12.23	31	188	2.58	23.94
34	Z5xp1	114	448	2.83	47.26	132	641	3.75	61.41
35	Z9sym	71	330	2.45	25.45	77	363	5.36	41.33

Table 1 gives the results comparing circuits using Timed Shannon Circuits versus those obtained from Synergy using logic optimization to minimize the area of the circuit with no regard for the delay. This experiment is used to determine the potential area penalty incurred by using TSC's. All of the data was derived after technology mapping to an industrial standard CMOS cell library for a 0.5 micron fabrication process. The starting point for both approaches is the given un-optimized description of the publicly available benchmark examples. The TSC circuits are derived within the SIS system after performing standard logic optimization (using the script.rugged optimization script) using the three step derivation described in Section 2.0. The column titled **In** and **Out** gives the number of inputs and outputs for the circuit; the column titled **Gate Count** gives the number of library gate instances in the circuit; the column titled **Area** is the total cell area excluding routing; the col-

umn titled **Delay** is the delay of the combinational logic in nano-seconds; the column titled **Power** is the power dissipation estimated using the simulation based power estimation approach described in Section 3.0.

There are several immediate observations that can be made from the data in Table 1:

- The area penalty incurred by the TSC implementation is significantly higher than that of area-optimized circuits from Synergy. This is reflected both by the Gate Count and Area columns on the majority of examples. There are two main explanations for the large increase in area using TSC's. First, the circuits using the TSC implementation are derived from BDD's. A BDD for each output is built in terms of the primary inputs of the circuit. On almost all the circuits the size of the BDD is significantly larger than the size of the optimized circuit in Synergy. This penalty remains reflected in the final mapped circuits. Second, the TSC circuits are derived within SIS, which appears to be substantially inferior in logic optimization quality than Synergy. As a basis for this deduction we have compared the area results reported in the earlier work on TSC's [1], where the area of the TSC circuits were compared against optimized circuits in SIS. In almost all cases the area-optimized circuits from SIS are larger than those of Synergy - note that although this data has been collected it is not shown in the table. The top graph in Figure 6 illustrates the comparison on all 35 examples.

- The power dissipation of the TSC implementation is less than that of the Synergy circuits on only 5 of the 32 circuits. Even on these 5 cases the reduction in power dissipation is relatively small. These results are in sharp contrast to the power dissipation comparison described previously in [1]. The explanation appears to mostly lie in the significant improvement in the area optimization provided by Synergy over SIS. However there are two notable issues. First, the power dissipation per unit area (or per gate) for the TSC circuits is substantially lower than the power dissipation per unit area in the Synergy circuits. For a typical example, on the circuit apex4, the power dissipation per unit area for the TSC implementation is 0.02 whereas it is 0.04 for the Synergy implementation. The bottom graph in Figure 6 illustrates the power consumed per unit area. On 21 of the 35 examples, the TSC circuits dissipate less power per unit area than the Synergy optimized circuits. On most of the remaining circuits the power dissipation per unit area is similar. This data indicates that the TSC circuits may only prove competitive if the area penalty incurred by deriving the circuits from BDD's is reduced substantially. This remains an on-going research problem. Second, we have also observed (data not reported in the table) that the amount of power dissipation due to glitches in the Synergy optimized circuits is a very small fraction of the total power dissipation. The range we observed on the circuits in the table varied from close to 0 to 10%, with an average of less than 5% of the total power due to glitches. The contribution of the power due to glitches was substantially higher for the circuits optimized in SIS and reported in [1].

Scope for improvement in the area of the TSC circuits may be envisaged in two ways:

- Better derivation of circuits from BDD's. One of the main drawbacks of the current prototype is the use of a single BDD for each output - the BDD for each circuit output is built in terms of the primary inputs of the circuit. An approach where a single large BDD is decomposed into a set of smaller cascaded BDD's has already been shown to be critically important in cycle-based logic simulation and may prove effective even for TSC's. This remains an open research problem. In addition to algorithms for BDD decomposition, modified enabling logic for the set of cascaded BDD's also has to be developed.

- Incorporation of the optimizations used by Synergy in the TSC derivation. The current prototype performs technology independent optimizations on the TSC using the logic optimization commands available in SIS; however, our experiments have demonstrated that the area optimizations in Synergy are substantially superior and their use in the TSC derivation may improve the resulting area.

The data in Table 2 compares the results of circuits optimized for delay in Synergy versus the same TSC circuits used in Table 1 - the only difference for the latter being that technology mapping was performed to satisfy delay constraints rather than area optimization constraints used for the first experiment. The data in this table has to be interpreted with greater caution because of the inability of the TSC implementations to match the timing of the Synergy circuits. Figure 7 provides a graphical comparison of the performance of the TSC and Syn-

ergy circuits. The power dissipated by the TSC circuits is significantly lower than the Synergy circuits for at least 11 of the circuits (*e.g. alu4, apex3, apex4, apex5, pdc, seq*). However with the exception of the *pdc* example, the TSC implementation fails to meet the specified timing constraints. In most of the cases the delay penalty of the TSC circuit is substantial. It remains unclear how the delay of the TSC's can be improved significantly enough to be competitive with Synergy. By its nature, the TSC trades-off delay for power consumption - to ensure a minimum amount of transition activity the evaluation of the circuit proceeds from the Enable signal towards the output in a serial fashion through the conditional buffering trees introduced by the third step of the TSC derivation (*c.f.* Section 2.0) to reduce the high fanout on primary inputs.

FIGURE 6. Power, area, and power per unit area comparisons for TSC versus Synergy - Area mode optimization.
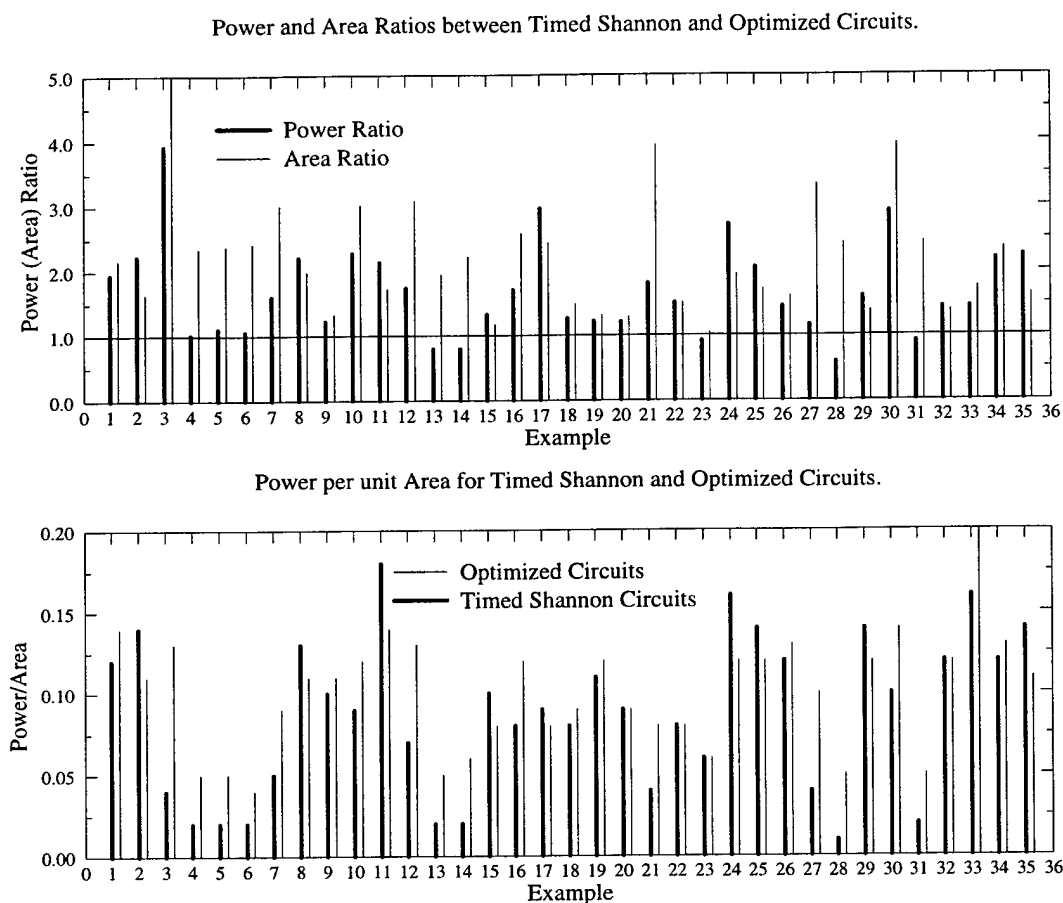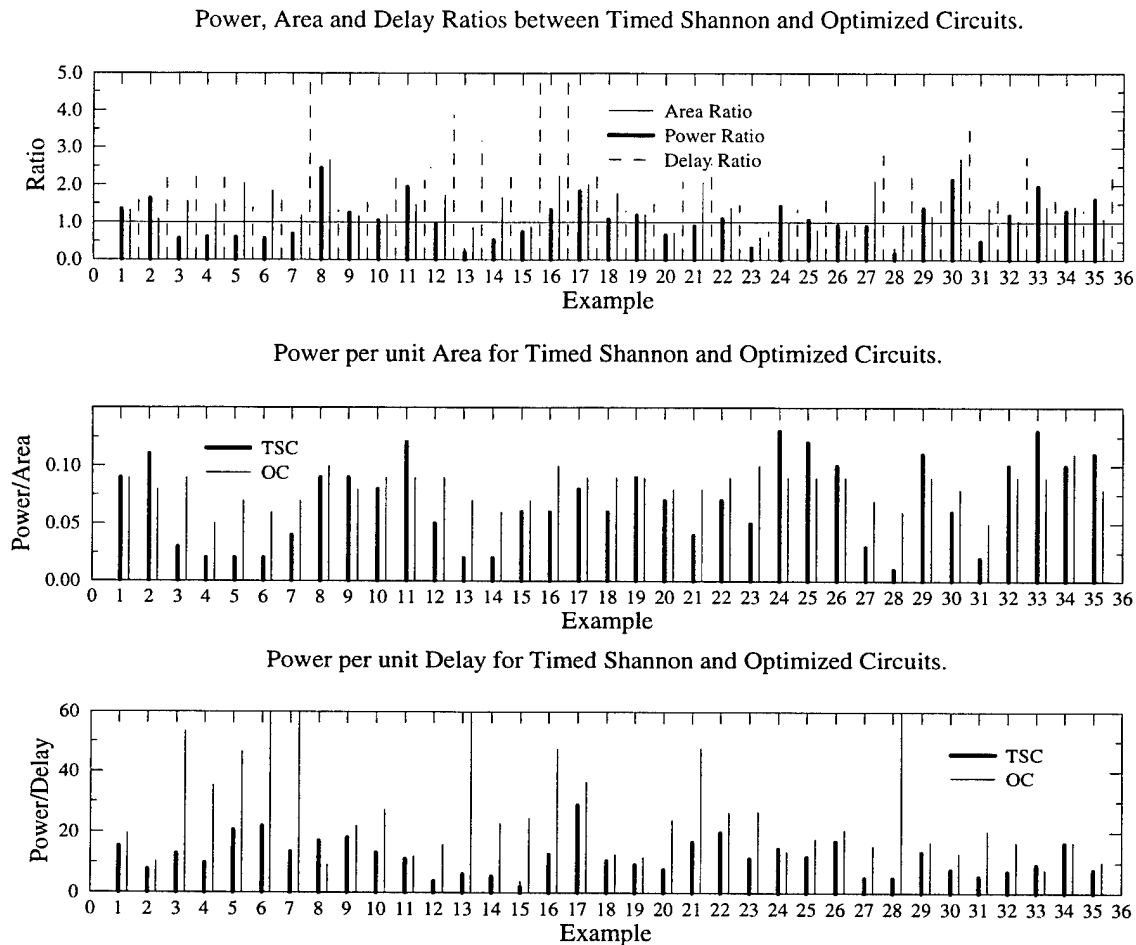

Power and Area Ratios between Timed Shannon and Optimized Circuits.

Power per unit Area for Timed Shannon and Optimized Circuits.

FIGURE 7.   Power, area, delay, power per unit area, and power per unit delay comparisons- Timing mode optimization.

Power, Area and Delay Ratios between Timed Shannon and Optimized Circuits.

Power per unit Area for Timed Shannon and Optimized Circuits.

Power per unit Delay for Timed Shannon and Optimized Circuits.

6.0   Related Work

In closely related work, we have explored the impact of low-power optimization under a trace-driven synthesis methodology [9]. Given a logic description of a digital circuit C and an expected trace of input vectors T, an implementation of C that optimizes a cost function under application of T is derived. This approach is effective in capturing and utilizing the correlations that exist between input signals on an application specific design. The idea is novel since it proposes synthesis and optimization at the logic level where the goal is to optimize the average case rather than the worst case for a chosen cost metric. The work reported in [9] focuses on the development of algorithms for trace driven optimization to minimize the switching power in multi-level networks. The technique is mainly applicable to the reduction of switching power within the combinational logic of finite state machines (FSM's). The average net power reduction (internal plus I/O power) obtained on a set of benchmark FSM's is 14%, while the average reduction in internal power is 25%. The primary result of the research in [9] is the demonstration that the I/O transition activity provides a dominating upper bound on the power reduction that can be achieved by combinational logic synthesis. The I/O power accounts for 20% up to

46% of the total power dissipation on the examples reported in [9]. As an example of this dominance, a power reduction of 42% on the internal gates of an example circuit realized only a net reduction of 25% since the I/O power accounts for 40% of the total power. I/O switching activity can only be changed by changing the sequential behavior of the circuit (*e.g.* by state encoding or latch retiming optimizations).

7.0 Conclusions

In this report we have described the results of an experiment to determine the feasibility of deploying Timed Shannon Circuit implementations in industrial setting. Analysis of the data from the experiment leads to the conclusion that the existing implementation of the Timed Shannon Circuit approach is not competitive with a commercial logic optimization program. Although the data collected is comprehensively negative on the surface with regard to the use of Timed Shannon Circuit technology there are two moderately promising observations that merit further research. First, it appears that a TSC implementation may be effective if the area overhead is reduced. Second, on a few circuits the TSC implementation could provide a useful power-area-delay trade-off in comparison to some of the implementation derived by Synergy. Unfortunately, both of these avenues currently are difficult open problems and it is infeasible to predict the duration of such an undertaking to explore these problems as well as the potential benefits that they may yield. In addition, work not directly related to Timed Shannon Circuits, but relevant nonetheless to determining the feasibility of low-power optimization for combinational logic circuits indicates that the limiting factor on power reduction is the I/O switching power which is determined only by the function of the circuit, not its implementation. Thus one may surmise that low-power optimization on combinational logic circuits is extremely limited in scope and is not a fruitful enough area of exploration for power reduction in digital electronic systems. From the published literature, far more substantial reduction in power dissipation is achieved at the architectural (or behavioral) level of system implementation as well as in the technology specific physical process domain [10].

References

[1] L. Lavagno, P. McGeer, A. Saldanha and A. Sangiovanni-Vincentelli. Timed Shannon Circuits: A Power-Efficient Design Style and Synthesis Tool. In *Proceedings of the 32nd Design Automation Conference*, pages 254-260, June 1995.

[2] M. Pedram. Power Minimization in IC Design: Principles and Applications. In *ACM Transactions on Design Automation of Electronic Systems*, pages 3-56, Jan. 1996.

[3] V. Simion. Interface Standards Activities & Translators. A Report for the GloMo Information Systems Program. Cadence Design Systems, Inc., Oct. 1996.

[4] S. M. Kang and Y. Leblebici. *CMOS Digital Integrated Circuits: Analysis and Design*. McGraw-Hill, New York, 1996.

[5] S. Yang. *Logic Synthesis and Optimization Benchmarks User Guide*. Microelectronics Center of North Carolina, Jan. 1991.

[6] R. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, and A. Wang. MIS: Multiple-Level Logic Optimization System. In *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, pages 1062-1081, Nov. 1987.

[7] R. Brayton, G Hachtel, C. McMullen, A. Sangiovanni-Vicentelli. *Logic Minimization Algorithms for VLSI Synthesis*. Kluwer Academic Publishers, 1984.

[8] J. Vasudevamurthy, and J. Rajski. A Method for Concurrent Decomposition and Factorization of Boolean Expressions. In *Proceedings of the International Conference on Computer-Aided Design*, pages 510-513, Nov. 1990.

References

[9] L. Carloni, P. McGeer, A. Saldanha and A. Sangiovanni-Vincentelli. Trace Driven Logic Synthesis - Application to Power Minimization. To appear In *Proceedings of the International Conference on Computer-Aided Design*, November 1997.

[10] A. Chandrakasan, M. Potkonjak, J. Rabaey, and R. W. Broderson. Hyper-LP: A system for power minimization using architectural transformations. In *Proceedings of the International Conference on Computer-Aided Design*, pages 300-303, November 1992.

PAGE INTENTIONALLY LEFT BLANK.